Grid Security: Requirements, Plans and Ongoing Efforts

2003 ACM Workshop on XML Security

George W. Johnson Center at George Mason University, Fairfax, VA
October 31, 2003

Frank Siebenlist

The Globus Alliance (www.globus.org)
Mathematics and Computer Science Division
Argonne National Laboratory
franks@mcs.anl.gov

The Globus Alliance

Making Grid computing a reality

- Close collaboration with real Grid projects in science and industry
- Development and promotion of standard Grid protocols (e.g. OGSA) to enable interoperability and shared infrastructure
- Development and promotion of standard Grid software APIs and SDKs to enable portability and code sharing
- The Globus Toolkit®: Open source, reference software base for building Grid infrastructure and applications
- Global Grid Forum: Development of standard protocols and APIs for Grid computing

Content

- What makes the Grid "The Grid"
  - Global Grid Forum, OGSA, Globus Toolkit
- What makes Grid Security "special"
  - Virtualization vs least privilege delegation
  - Outsourcing the "whole" policy admin
  - Retracing and reconciliation
  - Do dynamic accounts have an "identity"?
  - End-to-end is the goal
  - Securely moving service instances
- Standards, standards, standards, standards…and concerns
  - WS Security, Liberty Alliance, OASIS' SAML & XACML, W3C
- Conclusions

NEESgrid Earthquake Engineering Collaboratory

U.Nevada Reno

www.neesgrid.org

Remote Users (Faculty, Students, Practitioners)

Instrumented Structures and Sites

Laboratory Equipment

High-Performance Network(s)

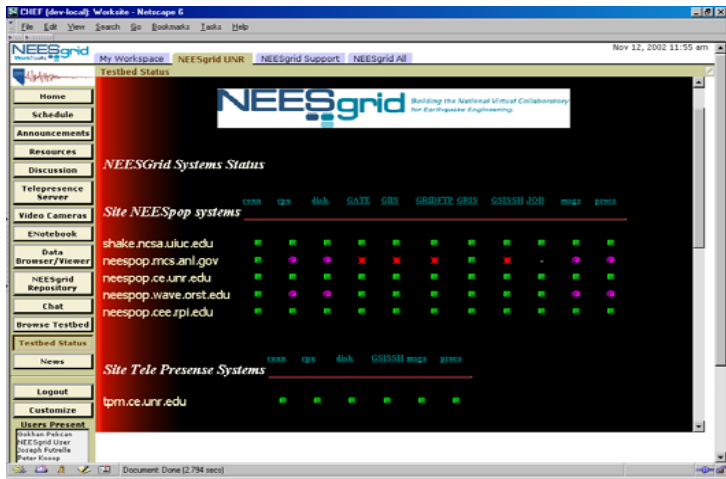
Field Equipment

Curated Data Repository

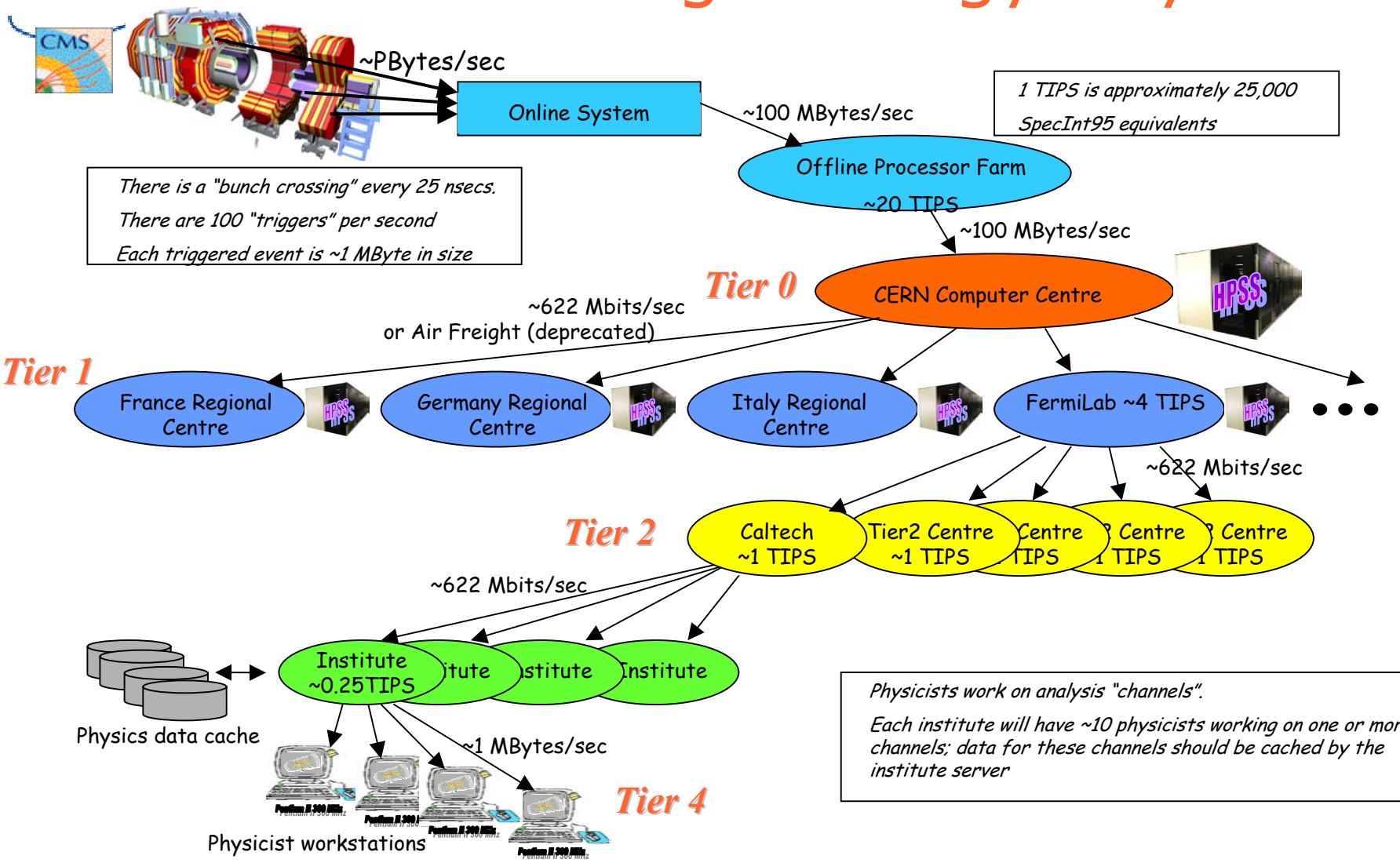
Global Connections (fully developed FY 2005 – FY 2014)

Laboratory Equipment (Faculty and Students)

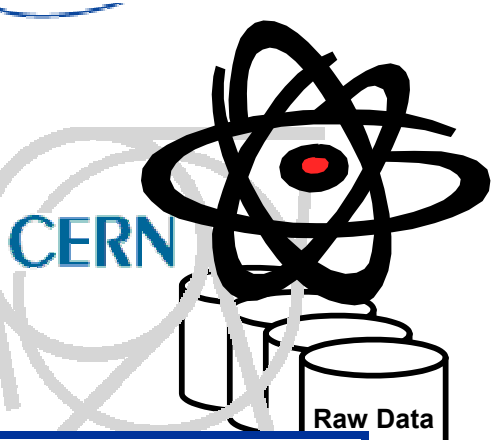
Remote Users: (K-12 Faculty and Students)



Data Grids for High Energy Physics



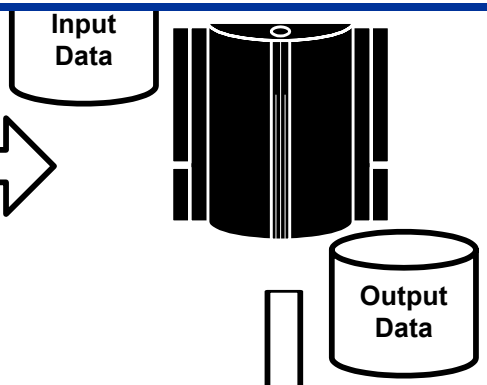
Grids: eXtreme Computing



PBytes

Gbits/s

TeraFlops



Mega $$$

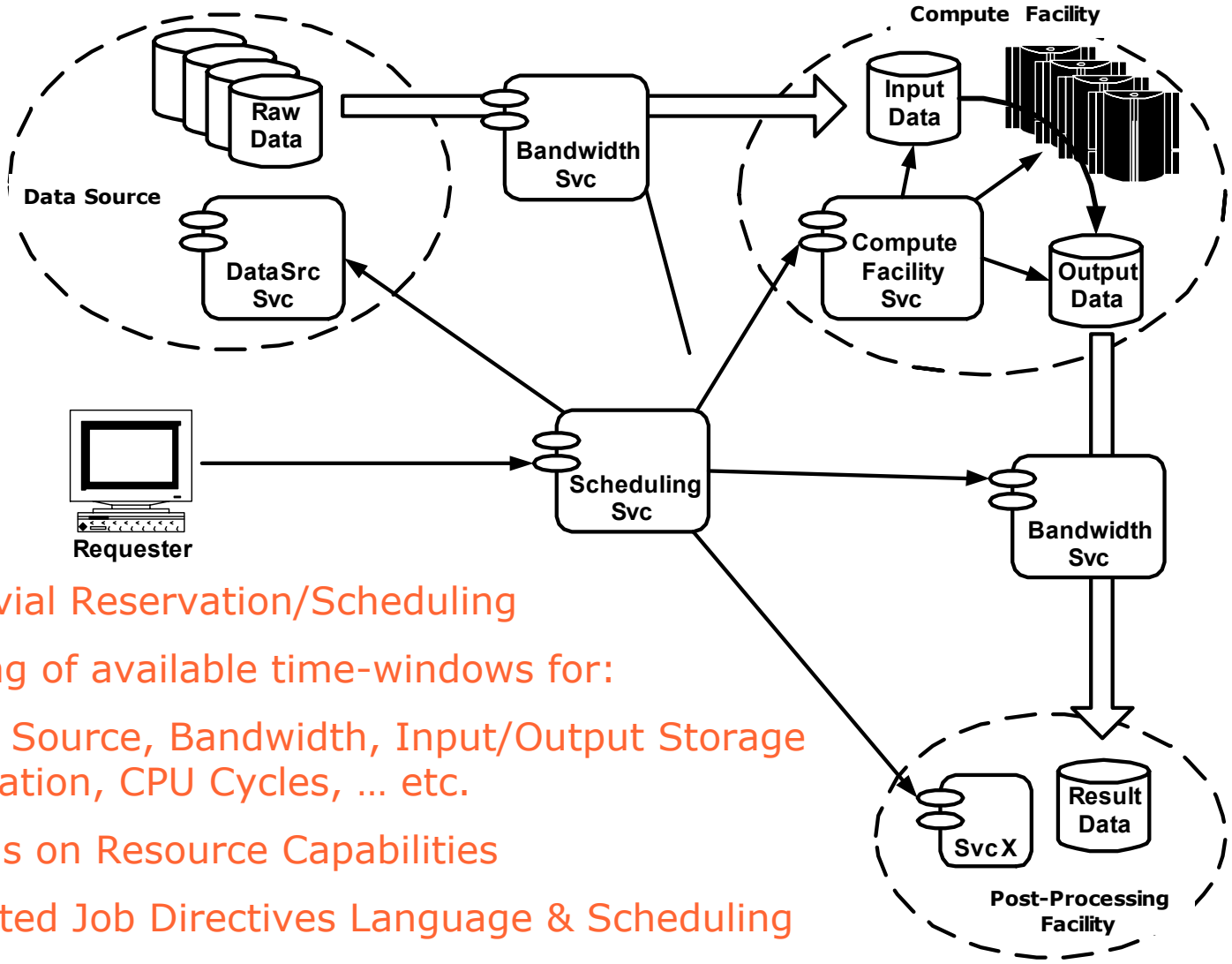
"Mad Scientists"

Cool Visualization

Grid Features

- **eXtreme requirements**
  - **Tera/Peta-Bytes**
  - **10-100 Gbits/sec**
  - **Giga/TeraFlops**
- High performance file transfer
  - Parallel Streaming
- Resource Sharing
  - Scheduling/Reservation
  - Job submission language
  - Non-trivial QoS
- Resource Virtualization
  - Publish/Discover Capabilities
  - Domain specific registries
  - Clustered/scavenging apps
  - Non-trivial QoS
- Data Virtualization
  - Abstraction of distributed data location
- Security
  - Virtual Organization=Bridge
  - Federate authN/authZ/policy
  - Delegation assertions
  - Non-trivial QoP negotiation
- Interoperability
  - Multi-platform
  - Open Source
  - Standardized
  - Vendor Support
- Robustness
  - Failure semantics from start
  - Soft-State management

Grids: Resource Sharing



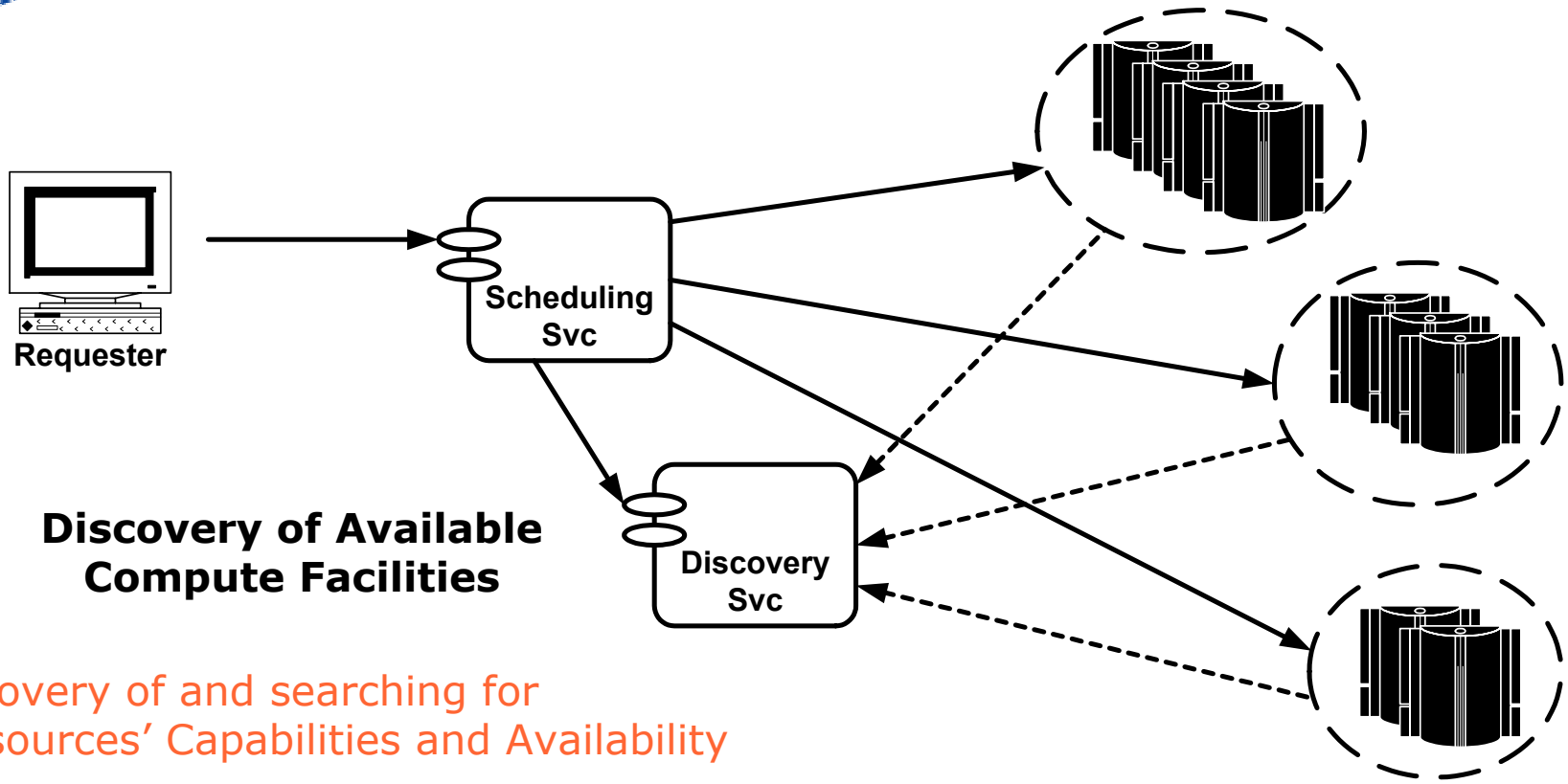
- Non-trivial Reservation/Scheduling
- Matching of available time-windows for:

  Data Source, Bandwidth, Input/Output Storage Allocation, CPU Cycles, … etc.
- Depends on Resource Capabilities
- Associated Job Directives Language & Scheduling
- It's all part of QoS Negotiation…

Grid Features

- eXtreme requirements
  - Tera/Peta-Bytes
  - 10-100 Gbits/sec
  - Giga/TeraFlops
- High performance file transfer
  - Parallel Streaming
- **Resource Sharing**
  - **Scheduling/Reservation**
  - **Job submission language**
  - **Non-trivial QoS**
- Resource Virtualization
  - Publish/Discover Capabilities
  - Domain specific registries
  - Clustered/scavenging apps
  - Non-trivial QoS
- Data Virtualization
  - Abstraction of distributed data location
- Security
  - Virtual Organization=Bridge
  - Federate authN/authZ/policy
  - Delegation assertions
  - Non-trivial QoP negotiation
- Interoperability
  - Multi-platform
  - Open Source
  - Standardized
  - Vendor Support
- Robustness
  - Failure semantics from start
  - Soft-State management

Grids: Resource Virtualization



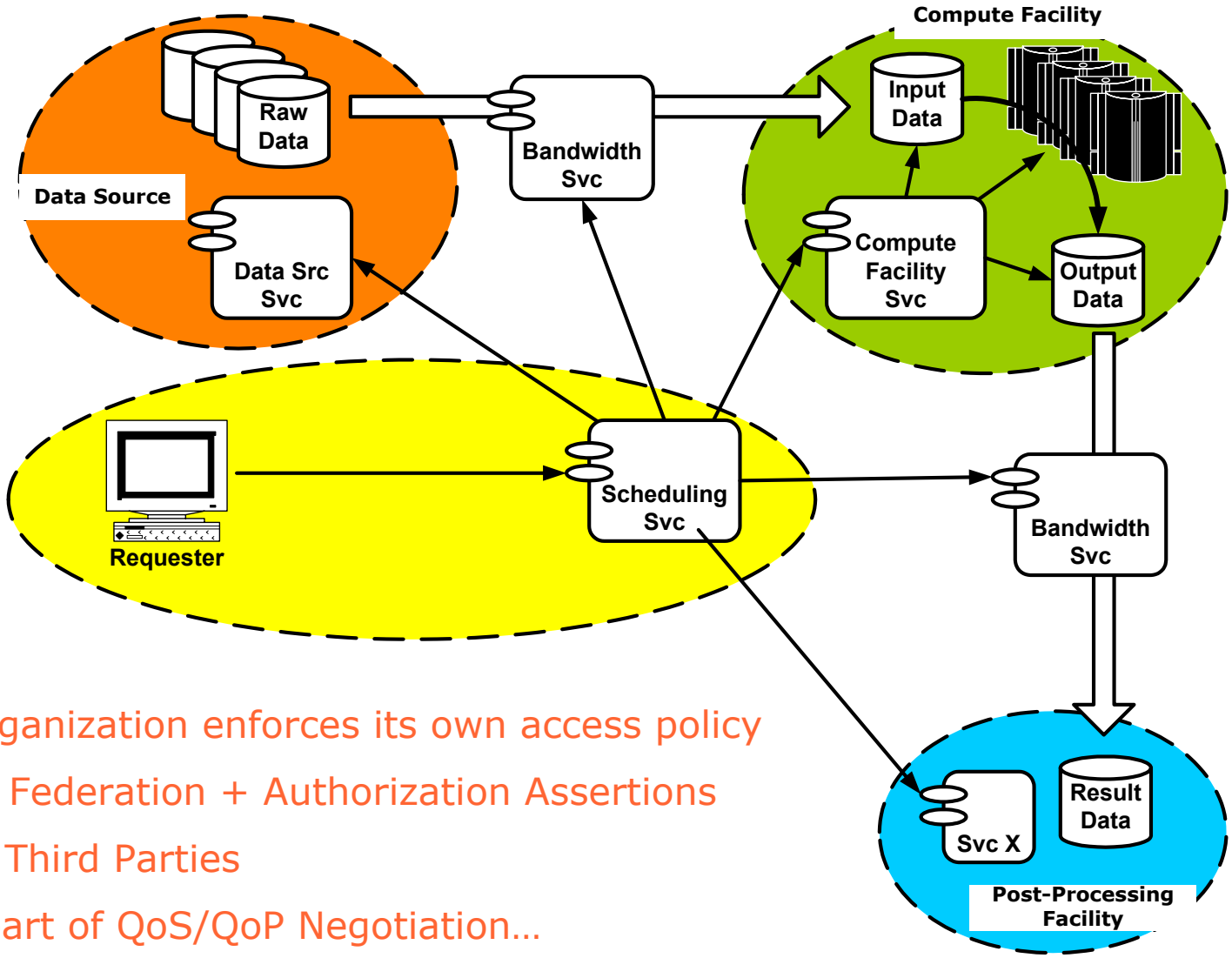
- Discovery of and searching for Resources' Capabilities and Availability
- Resource Capabilities:

  Amount of RAM/Storage/MFLOPS, # of CPUs, max. bandwidth,… etc.
- Use of actual Resources is "Virtualized"
- It's all part of QoS Negotiation…

Grid Features

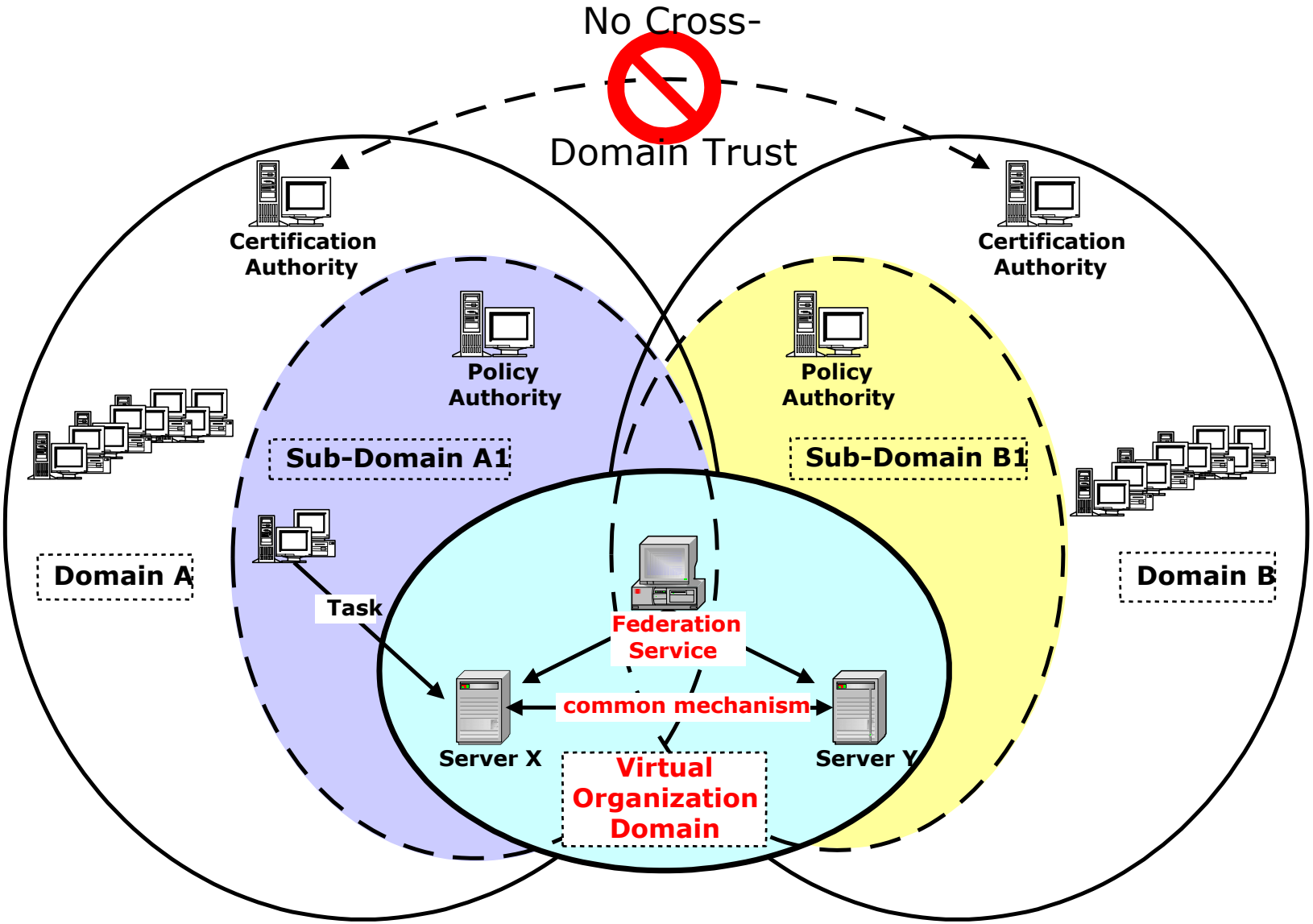
- eXtreme requirements
  - Tera/Peta-Bytes
  - 10-100 Gbits/sec
  - Giga/TeraFlops
- High performance file transfer
  - Parallel Streaming
- Resource Sharing
  - Scheduling/Reservation
  - Job submission language
  - Non-trivial QoS
- **Resource Virtualization**
  - **Publish/Discover Capabilities**
  - **Domain specific registries**
  - **Clustered/scavenging apps**
  - **Non-trivial QoS**
- Data Virtualization
  - Abstraction of distributed data location
- Security
  - Virtual Organization=Bridge
  - Federate authN/authZ/policy
  - Delegation assertions
  - Non-trivial QoP negotiation
- Interoperability
  - Multi-platform
  - Open Source
  - Standardized
  - Vendor Support
- Robustness
  - Failure semantics from start
  - Soft-State management

Grids: Multiple Independent Orgs



- Each Organization enforces its own access policy
- Identity Federation + Authorization Assertions
- Trusted Third Parties
- It's all part of QoS/QoP Negotiation…

Grid Solution: Use Virtual Organization as Bridge



Grid Features

- eXtreme requirements
  - Tera/Peta-Bytes
  - 10-100 Gbits/sec
  - Giga/TeraFlops
- High performance file transfer
  - Parallel Streaming
- Resource Sharing
  - Scheduling/Reservation
  - Job submission language
  - Non-trivial QoS
- Resource Virtualization
  - Publish/Discover Capabilities
  - Domain specific registries
  - Clustered/scavenging apps
  - Non-trivial QoS
- Data Virtualization
  - Abstraction of distributed data location
- **Security**
  - **Virtual Organization=Bridge**
  - **Federate authN/authZ/policy**
  - **Delegation assertions**
  - **Non-trivial QoP negotiation**
- Interoperability
  - Multi-platform
  - Open Source
  - Standardized
  - Vendor Support
- Robustness
  - Failure semantics from start
  - Soft-State management

Grid Features

- eXtreme requirements
  - Tera/Peta-Bytes
  - 10-100 Gbits/sec
  - Giga/TeraFlops
- High performance file transfer
  - Parallel Streaming
- Resource Sharing
  - Scheduling/Reservation
  - Job submission language
  - Non-trivial QoS
- Resource Virtualization
  - Publish/Discover Capabilities
  - Domain specific registries
  - Clustered/scavenging apps
  - Non-trivial QoS
- Data Virtualization
  - Abstraction of distributed data location
- Security
  - Virtual Organization=Bridge
  - Federate authN/authZ/policy
  - Delegation assertions
  - Non-trivial QoP negotiation
- Interoperability
  - Multi-platform
  - Open Source
  - Standardized
  - Vendor Support
- Robustness
  - Failure semantics from start
  - Soft-State management

What is a Grid?

- We believe there are three key criteria:
  - Coordinates resources that are not subject to centralized control …
  - using standard, open, general-purpose protocols and interfaces …
  - to deliver non-trivial qualities of service.
- What is not a Grid?
  - A cluster, a network attached storage device, a scientific instrument, a network, etc.
  - Each is an important component of a Grid, but by itself does not constitute a Grid

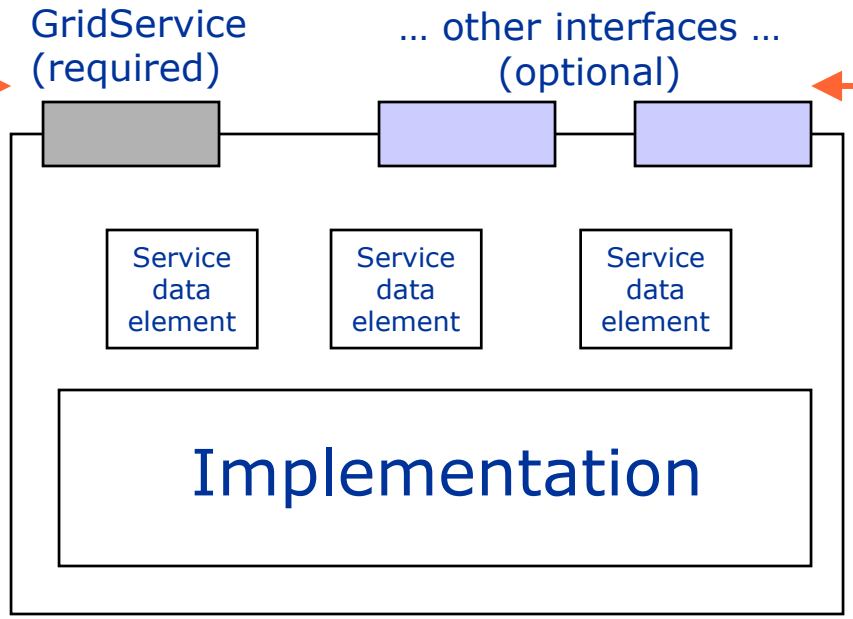
The Grid Service = Interfaces/Behaviors + Service Data

Open Grid Services Architecture

(OGSA = WebServices on Steroids)

Service data access
Explicit destruction
Soft-state lifetime

Support for stateful services



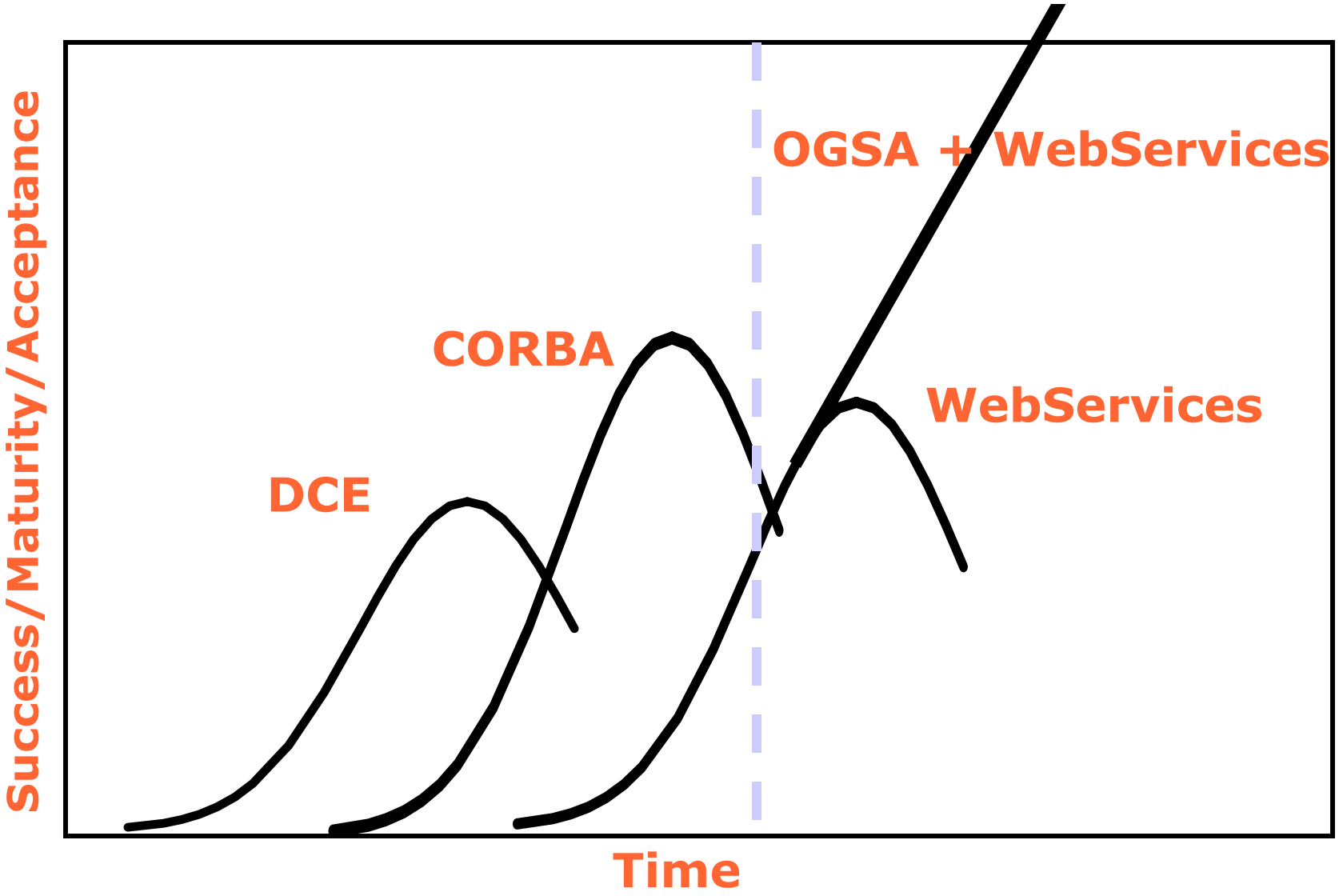
Standard:
- Notification
- Authorization
- Service creation
- Service registry
- Manageability
- Concurrency

+ application-specific interfaces

Binding properties:
- Reliable invocation
- Authentication

Hosting environment/runtime
("C", J2EE, .NET, …)

Silver Bullet Hype-Curve…



OGSA Security

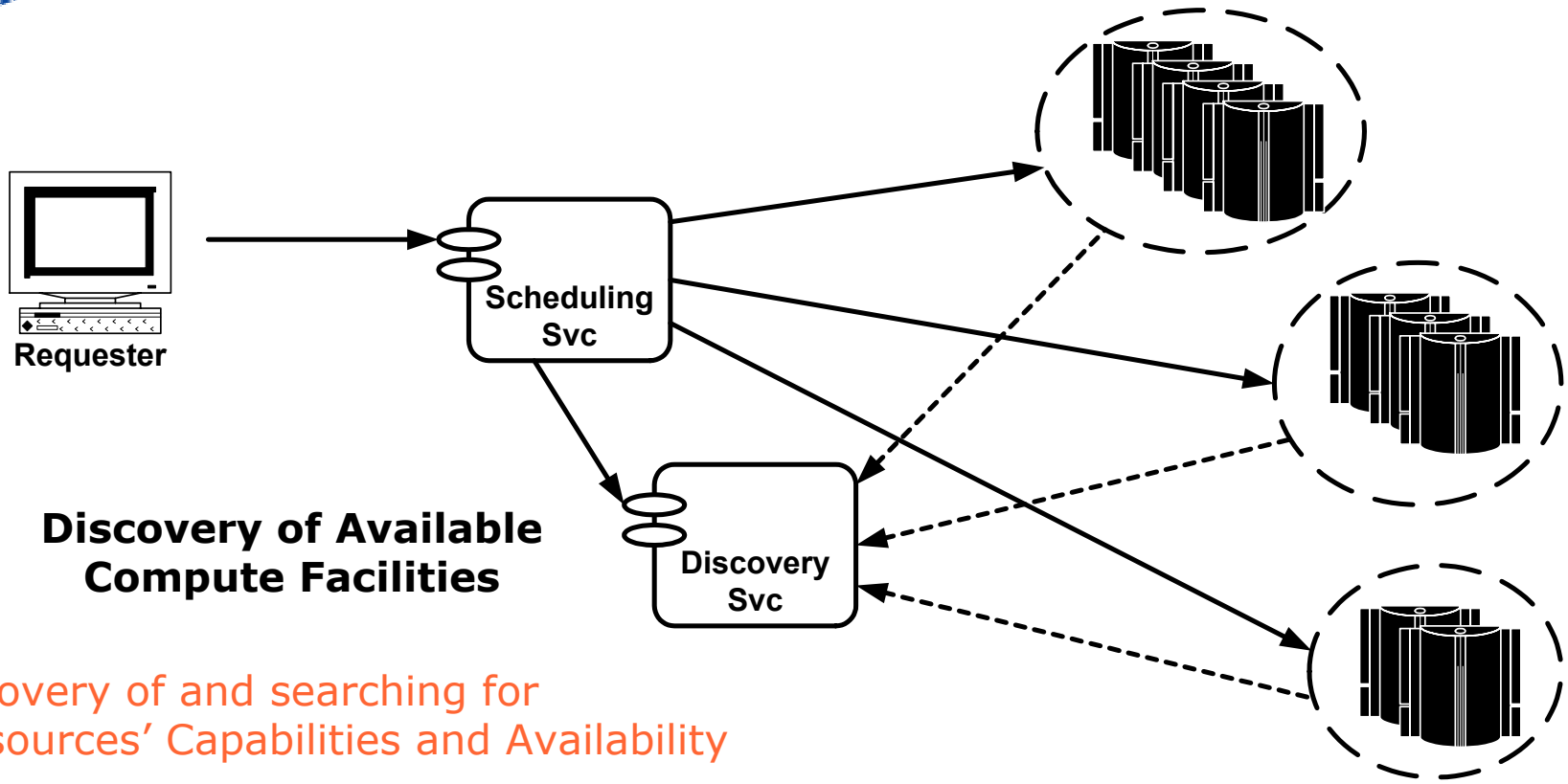
- Leverage existing/emerging WS security standards
- WS-Security/Policy/Trust/Federation/ Authorization/SecureConversation/Privacy
- XKMS, XML-Signature/Encryption, SAML, XACML, XrML

- But…
  - Need to OGSA'fy
  - Need to define Profile/Mechanisms
  - Need to define Naming conventions
  - Need to address late/missing specs
  - Support for delegation, transient services

What makes Grid Security "special"?

- Virtualization vs least privilege delegation
- Outsourcing the "whole" policy admin
- Retracing and reconciliation
- Do dynamic accounts have an "identity"?
- End-to-end is the goal
- Securely moving service instances

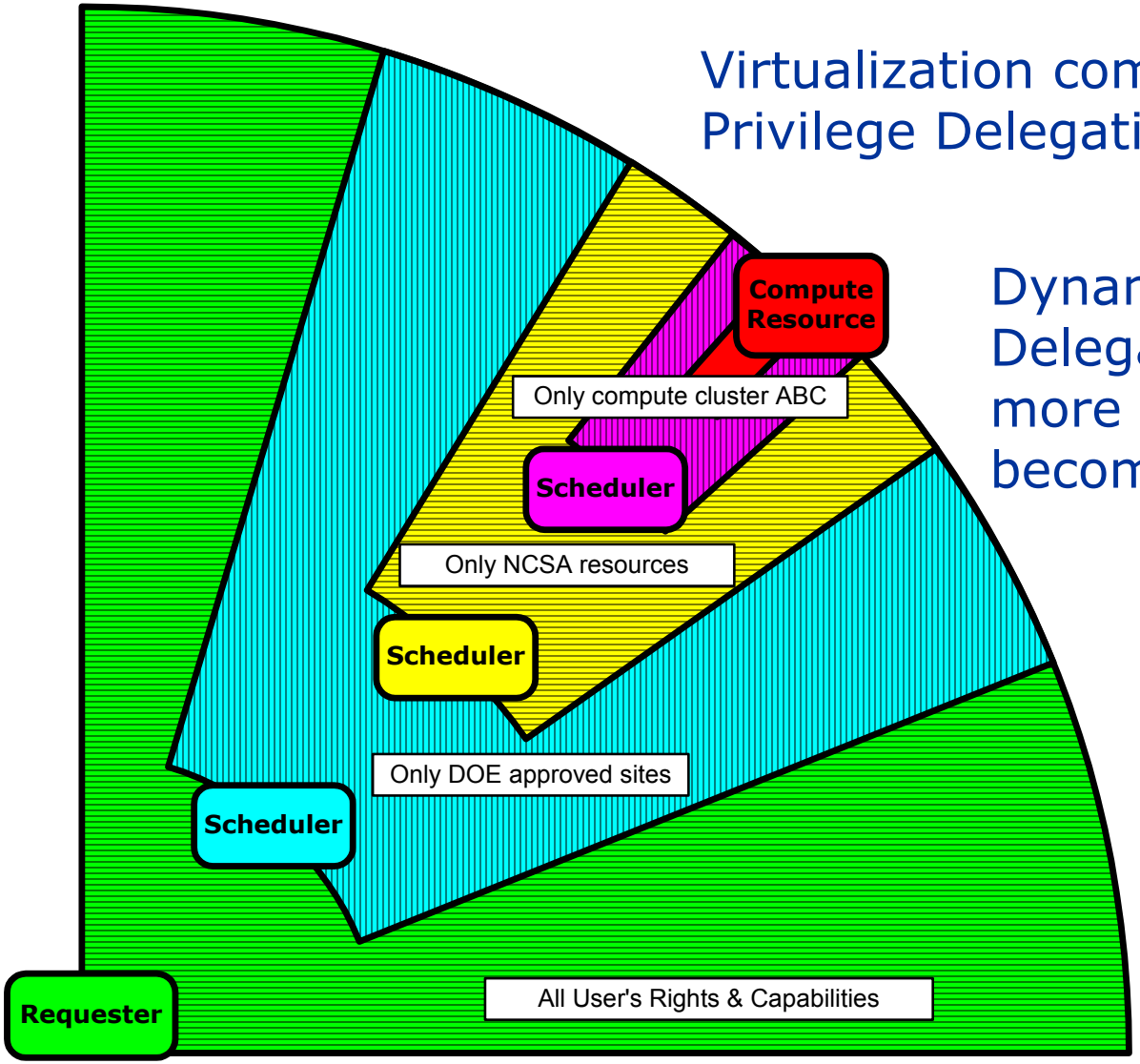
Grids: Resource Virtualization



- Discovery of and searching for Resources' Capabilities and Availability
- Resource Capabilities:

  Amount of RAM/Storage/MFLOPS, # of CPUs, max. bandwidth,… etc.
- Use of actual Resources is "Virtualized"
- It's all part of QoS Negotiation…

Propagation of Requester's Rights through Job Scheduling and Submission Process



Virtualization complicates Least Privilege Delegation of Rights

Dynamically limit the Delegated Rights more as Job specifics become clear

Trust parties downstream to limit rights for you… or let them come back with job specifics such that you can limit them

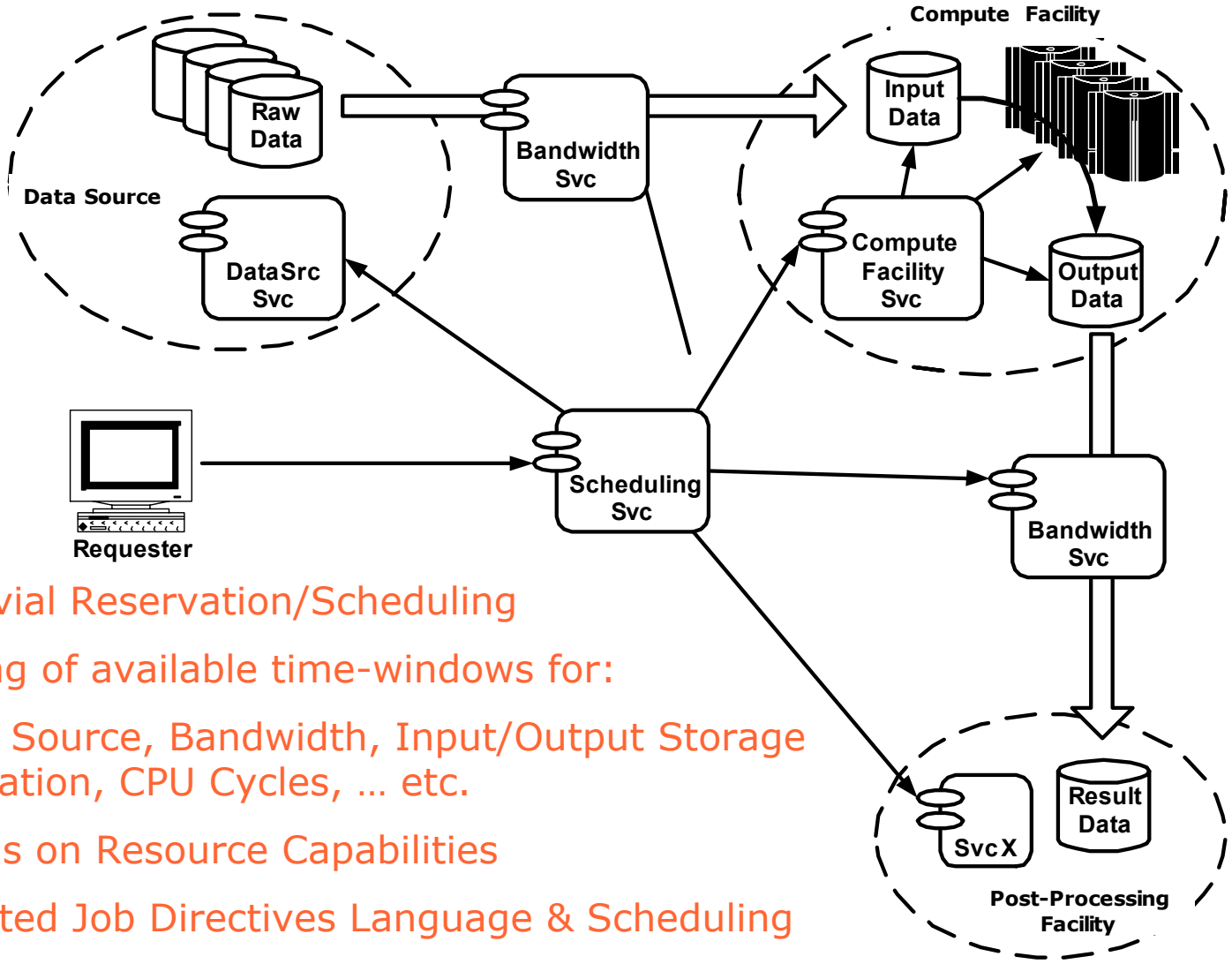
Delegation of Rights (1)

- Services "work on behalf of you"
  - Either explicitly or implicitly
- Services work on behalf of other services that work on behalf of you…
- Services need (a subset of) your rights
- Services are not under your control and are not even under your domain's control
- You will need a lot of "trust" … and the tools to limit the rights that go with your job
  - "I give that service the rights to represent me **only** for a specific set of operations on a specific set of resources"
  - "Furthermore, I give that service the rights to delegate a subset of those rights to other services"

Delegation of Rights (2)

- Need a standardized language to express and exchange authorization assertions
- XACML TC is adding delegation of rights features to 2.0
  - Learn from KeyNote, Delegation Logic, SPKI, etc.
  - XACML may be an "authorization assembler language"
- SAML Assertion may provide for signed envelope for XACML policy statement
- GGF's OGSA-Authorization WG may adopt…
- Need to tie closely in with Job description, scheduling and execution languages
  - Each has their own WG at GGF

Grids: Resource Sharing



- Non-trivial Reservation/Scheduling
- Matching of available time-windows for:

  Data Source, Bandwidth, Input/Output Storage Allocation, CPU Cycles, … etc.
- Depends on Resource Capabilities
- Associated Job Directives Language & Scheduling
- It's all part of QoS Negotiation…

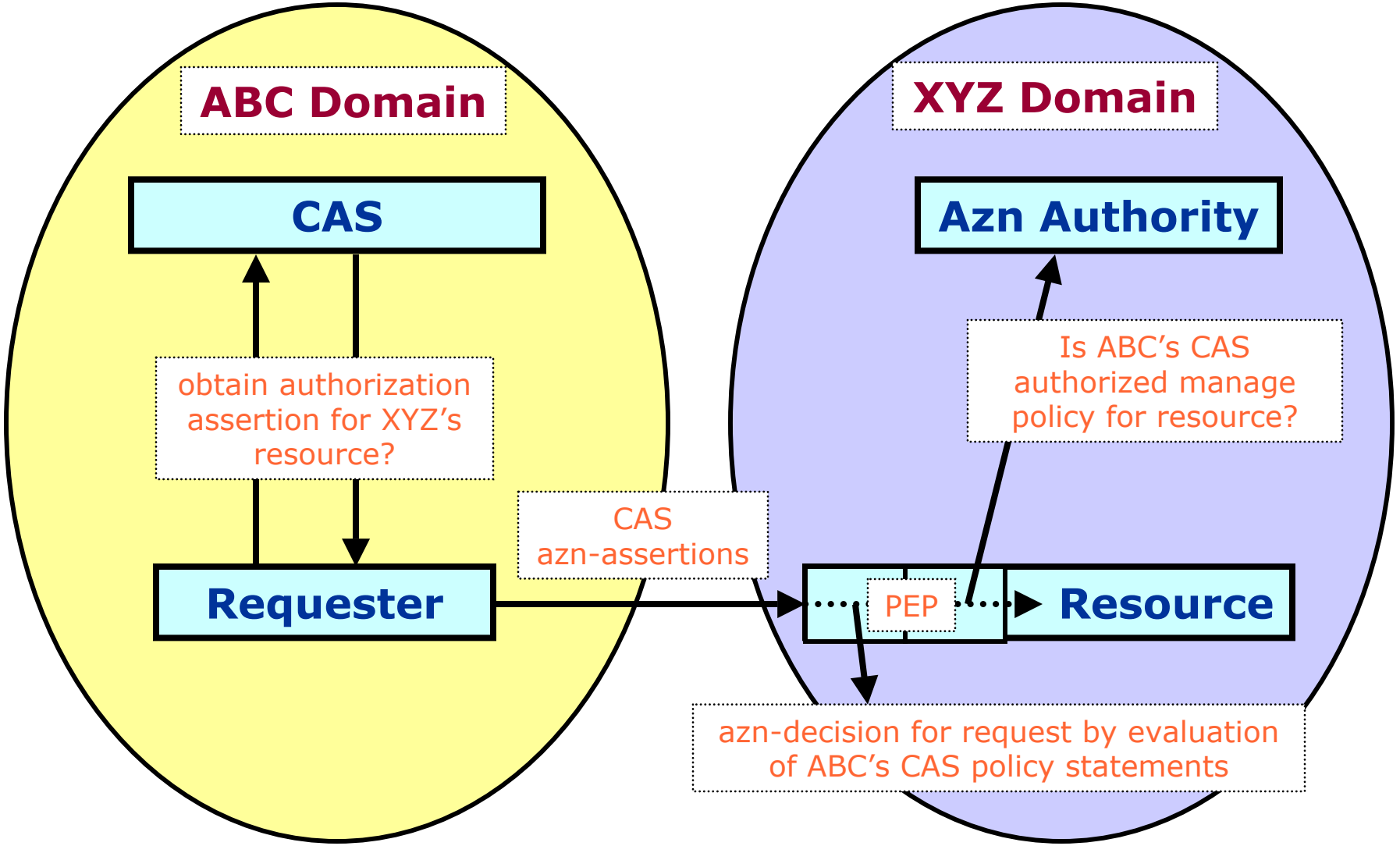
Job Scheduling and Authorization

- At each stage, the Job components and processing requests are subject to the local access control policy
- It can be expensive ($$$), if a job has to be aborted halfway because of authorization policy violation
- Authorization policy may have to be taken into consideration by the Scheduler
- Risk assessment:
  azn-policy exposure versus potential monetary loss
- Requirement for sharing of authorization policy
  - Integration of access control policy in scheduler/broker's scenarios and negotiations
- GGF's GRAAP WG and ws-agreement spec
  - dependencies on ws-policy-* and possibly xacml

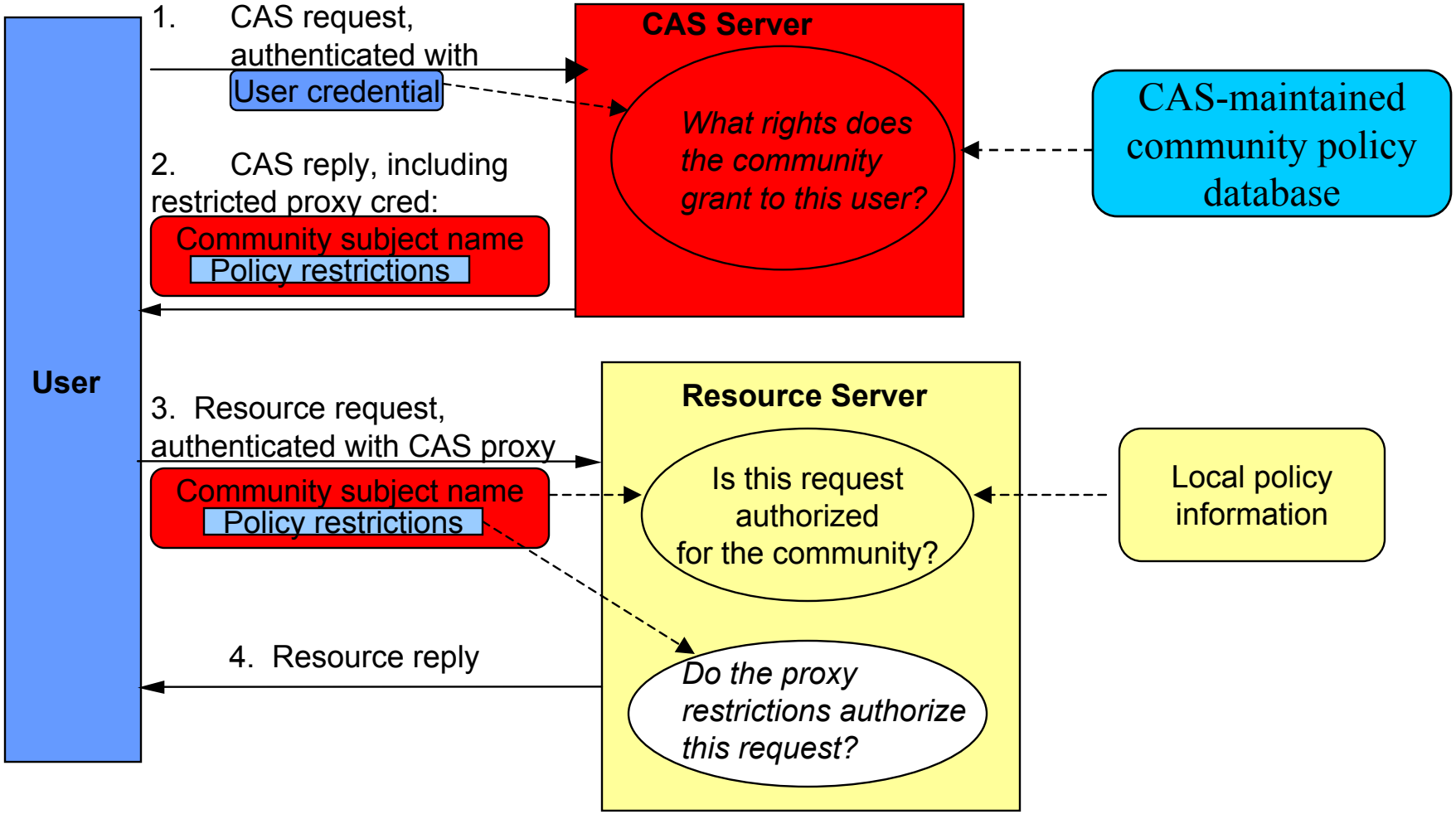
Outsourced Authorization Policy Admin

- Resource owner doesn't "know" foreign users and doesn't know details of resource usage
  - And doesn't want to know – a burden
- Agreement with foreign domain to outsource access control policy
- Different flavors:
  - Limited access to local policy admin tools
  - Outsource limited attribute assignments
  - Call-out to foreign AuthorizationDecision Service
  - Locally evaluate foreign policy statements
- In all cases, locally defined policy overrides
  - Local policy sets outer bounds

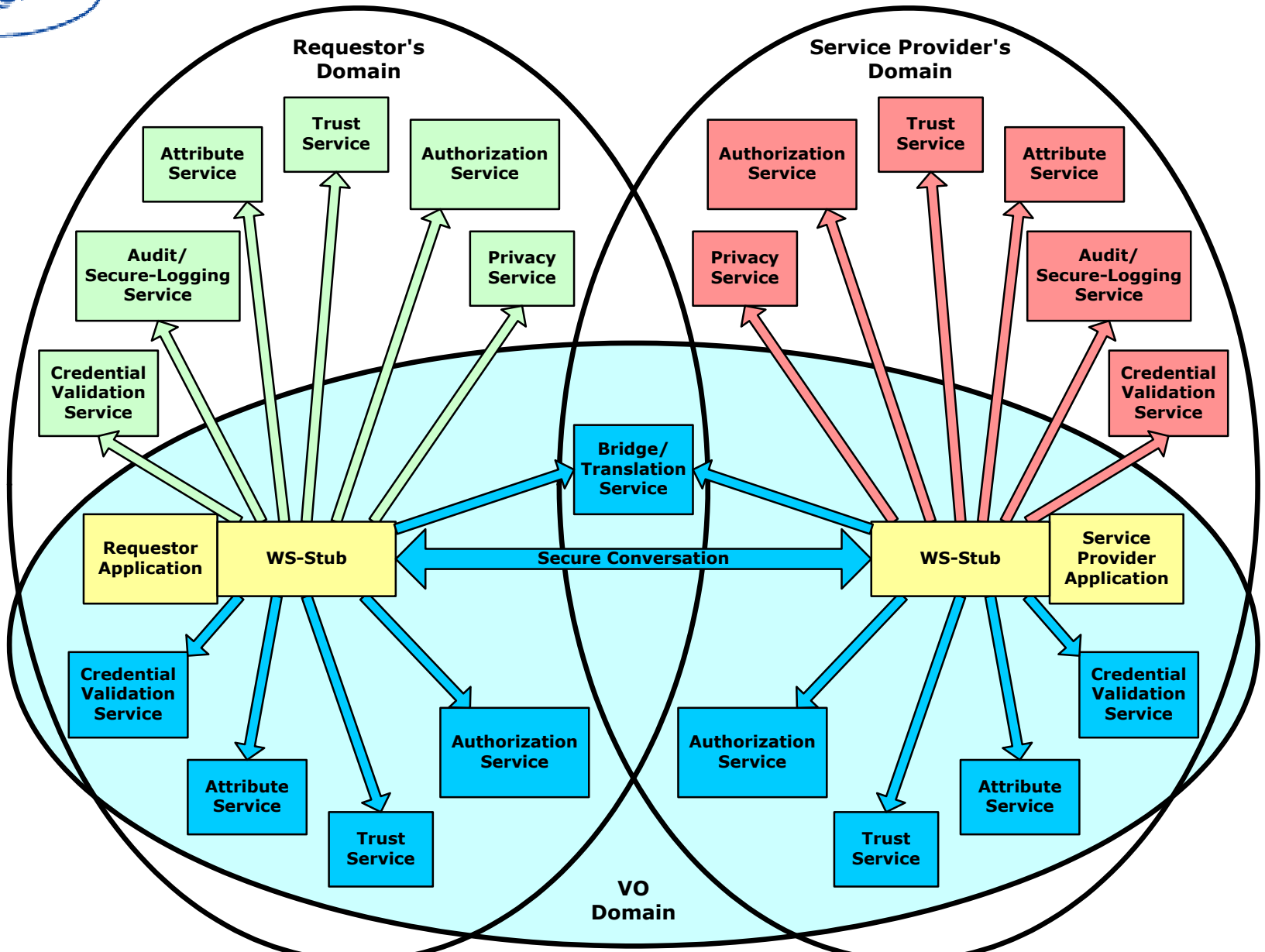
Community Authorization Service (CAS)



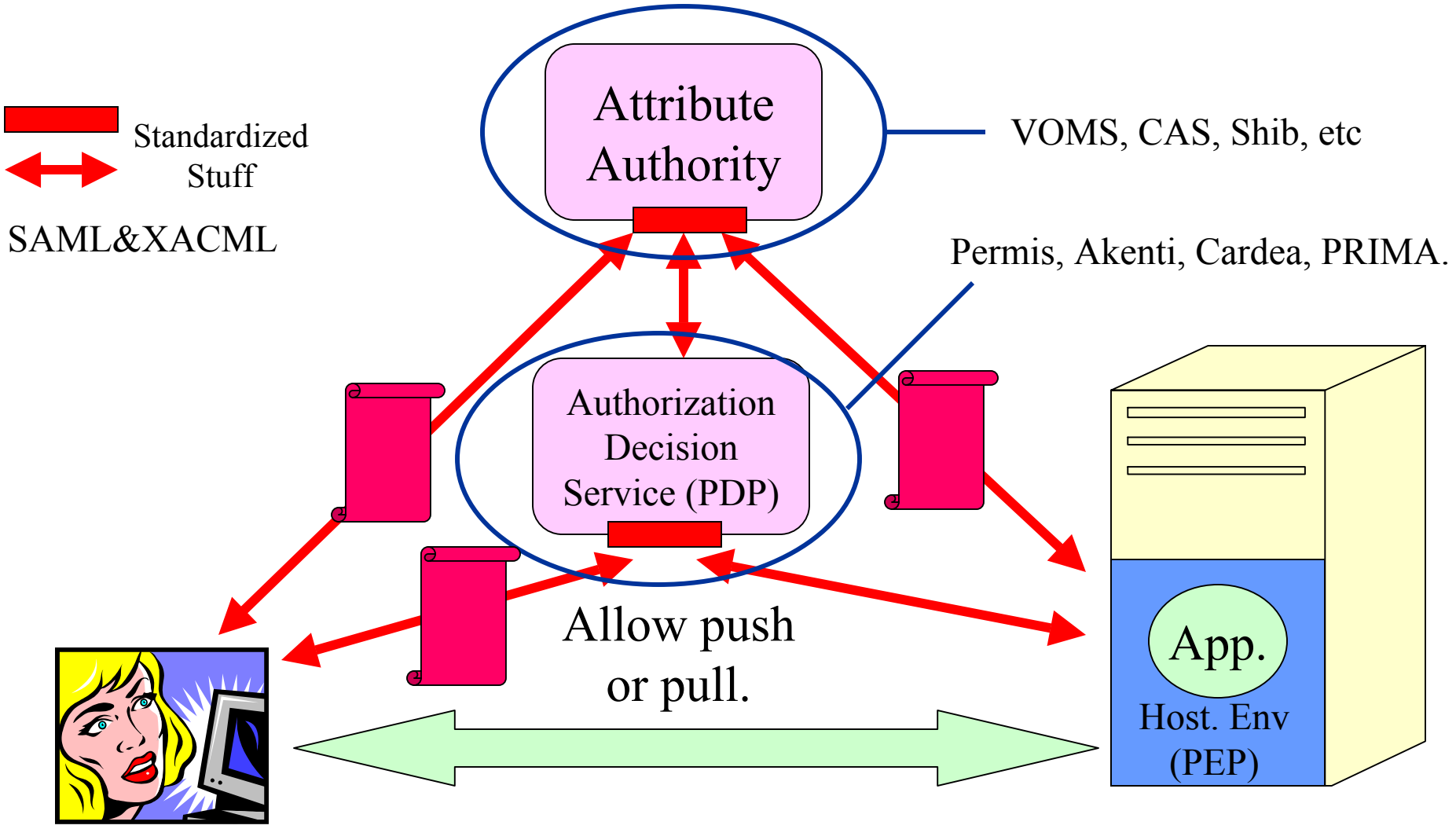
A Typical CAS Request



OGSA Security Services



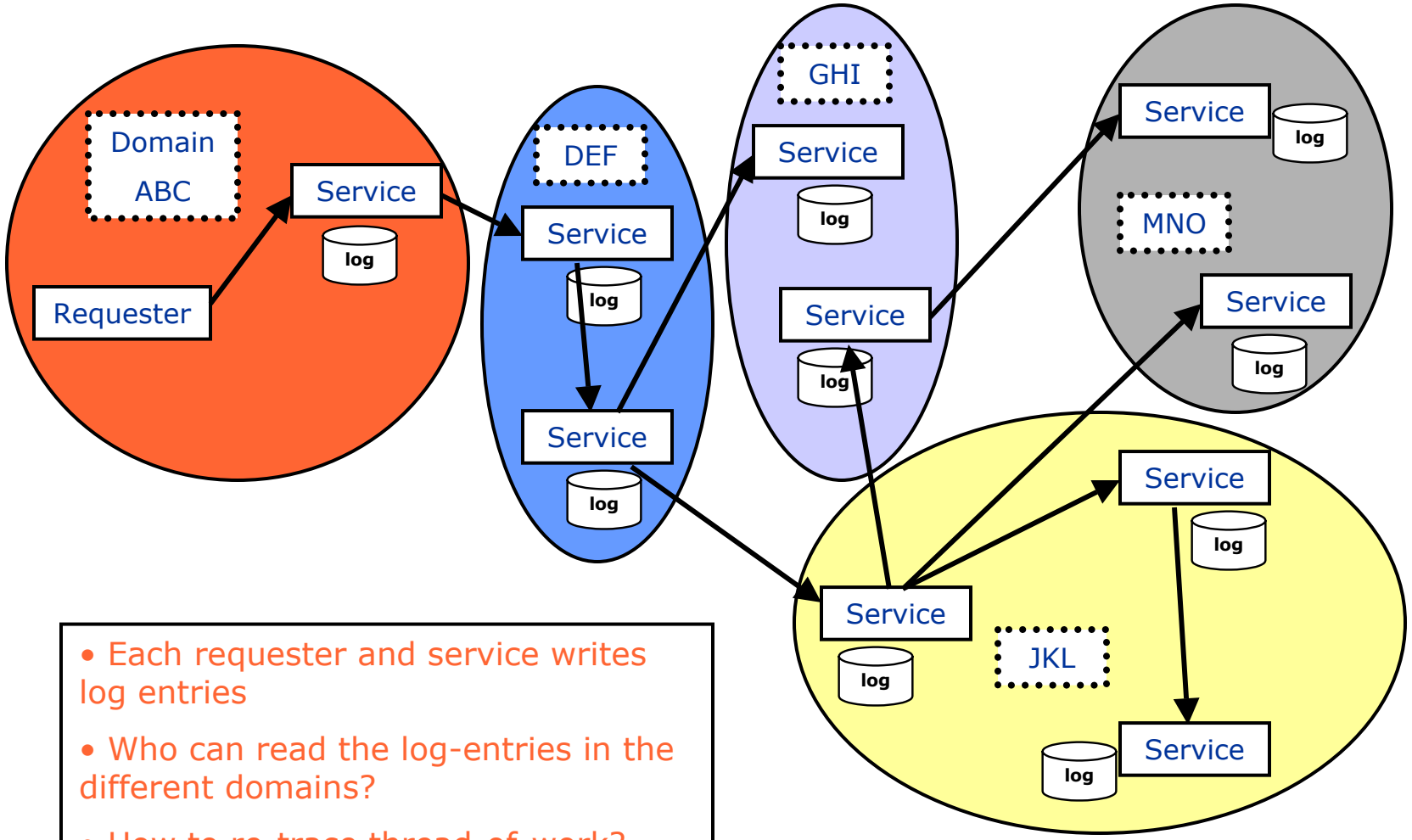
OGSA-Authz-WG Goals



Logging: need for keeping records…

- We will always delegate too many rights, and partly work on good faith, and partly on the ability to check after the fact.
  - Unable to define the transactions narrow enough
  - Maybe too expensive or impractical/impossible
  - "Real World" has many example
- We need to rely on secure logging and audit to ensure policy compliance and ability to reconcile.
- Unless we can work on a better world where we can just trust each other…
  - No working group at GGF yet ;-)

Distributed Logging in the Grid

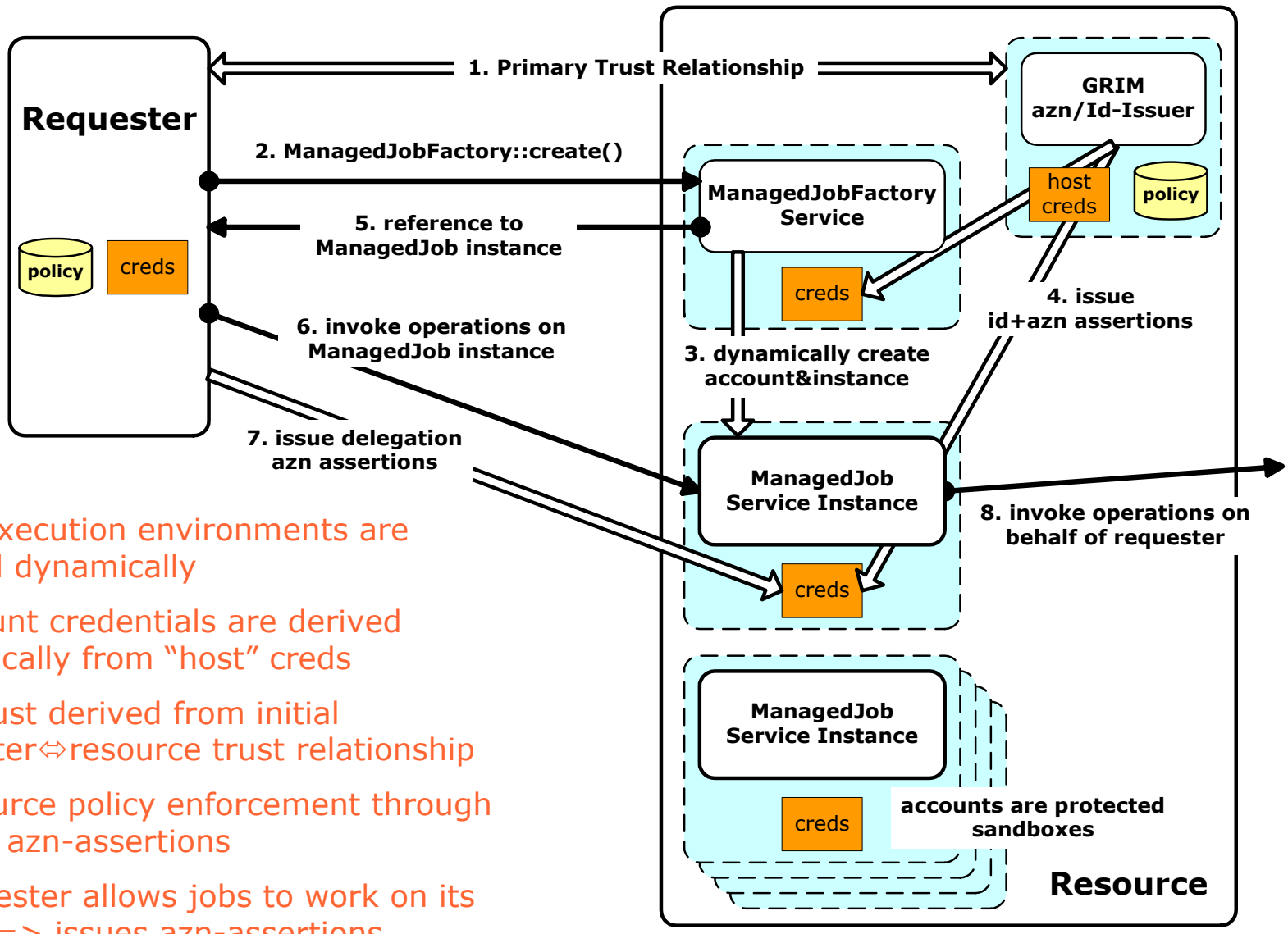


- Each requester and service writes log entries
- Who can read the log-entries in the different domains?
- How to re-trace thread-of-work?

Distributed Secure Logging

- Workflow paths may cross many administrative domains with different policies and technologies.
- Suppose we can solve interoperability, log entry format, correlation and tracing, interface standardization, etc., etc…,
  we will have very complicated access control policy challenge to "see" the log-entries
  - Separate kind of access policy if law enforcement is involved
  - Some domains/countries may have the legal requirements that the user must be able to "see" her/his associated entries…
- Start of logging service discussions in GGF's OGSA-WG
  - Very early stage … maybe BOF next GGF

GT3's Resource Management

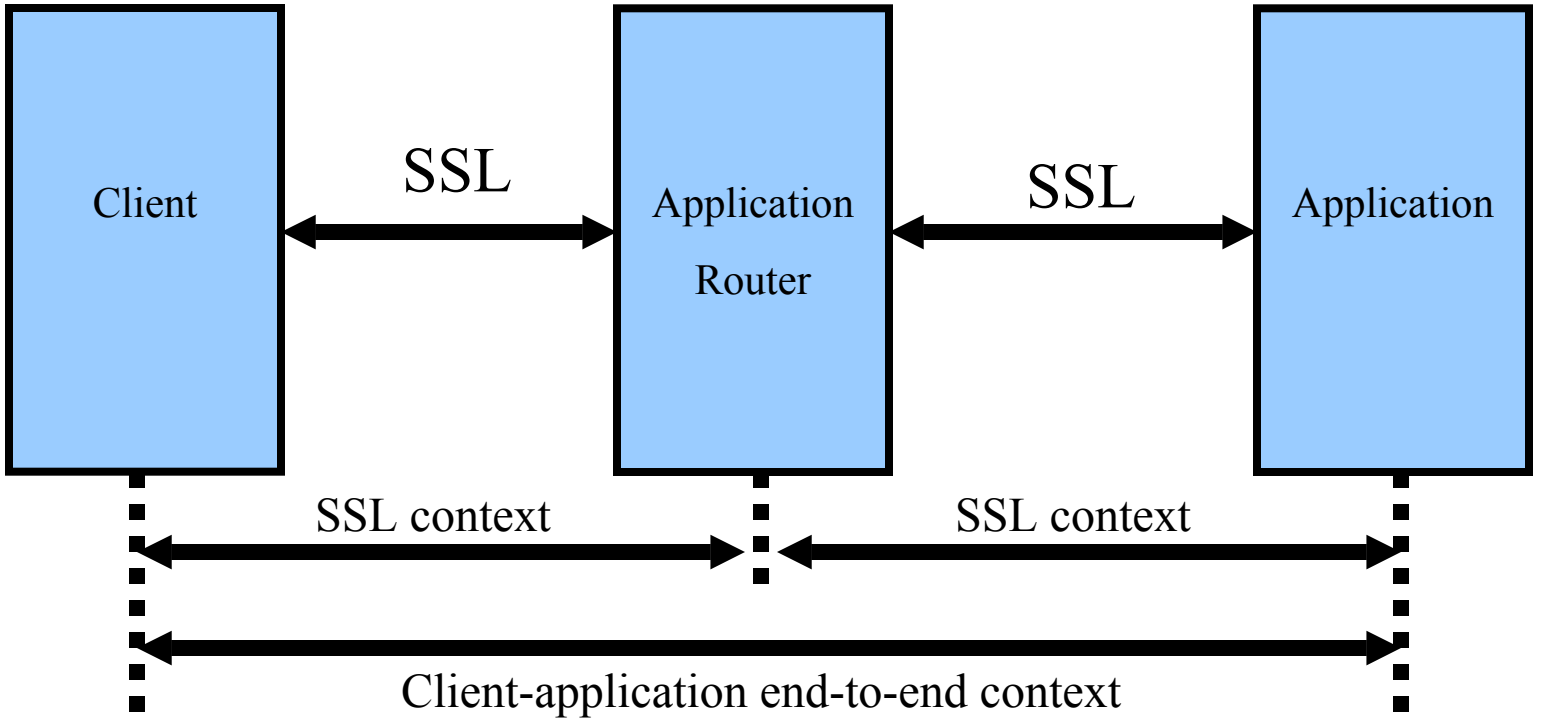


- Job execution environments are created dynamically
- Account credentials are derived dynamically from "host" creds
- All trust derived from initial requester⇔resource trust relationship
- Resource policy enforcement through GRIM's azn-assertions
- Requester allows jobs to work on its behalf => issues azn-assertions

Dynamic Resource Management

- Dynamic account/sandbox creation
  - X.509 identity registration procedure doesn't work…
  - Identity assertion not very useful…
- Newly created key pair are "the" identity creds
- Currently use proxy-certs to issue azn-assertions
  - GRIM asserts that requester can be trusted by account
  - GRIM asserts account can be trusted by requester
  - Requester asserts account can work on behalf of requester
- Future: XACML policy statements wrapped in SAML authorization assertions on bare keys issued by more permanent identities like host-identity and requester
- Leverage on GGF's OGSA-Authorization WG work

Transport vs Message Protection



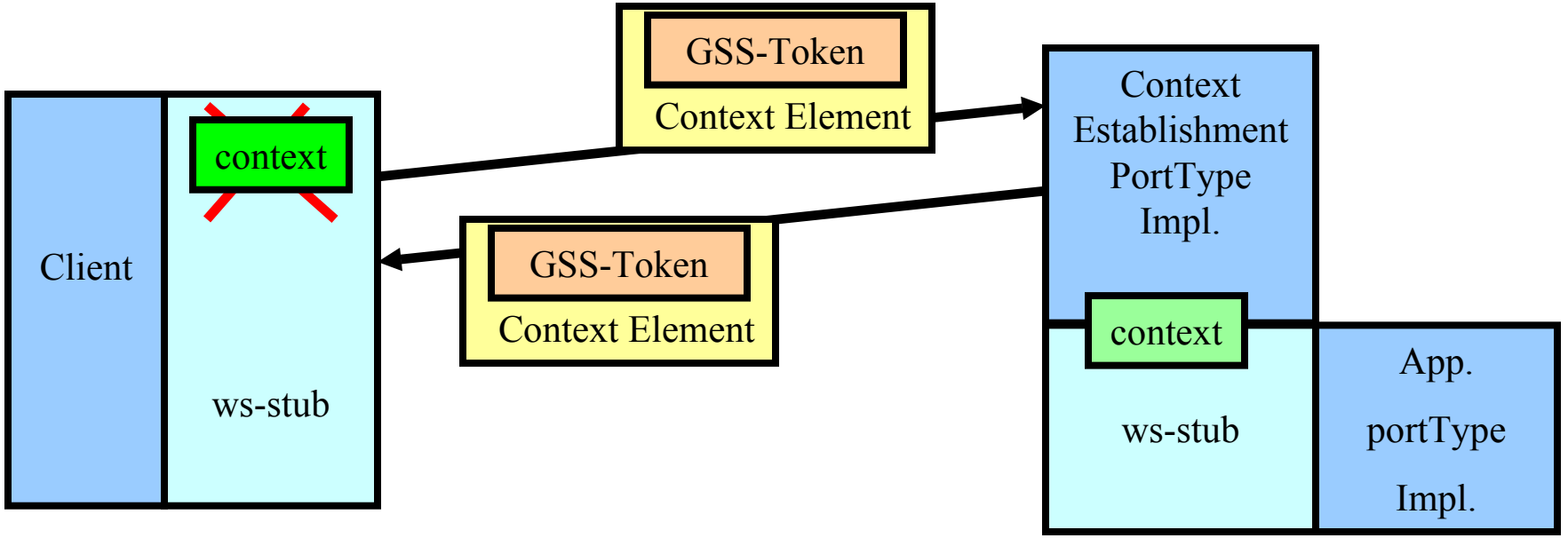
- **SSL Security Context determined by endpoints of socket connection**

  => Application Router becomes part of Trust Chain

- **Message level protection => end-to-end client-app security context**

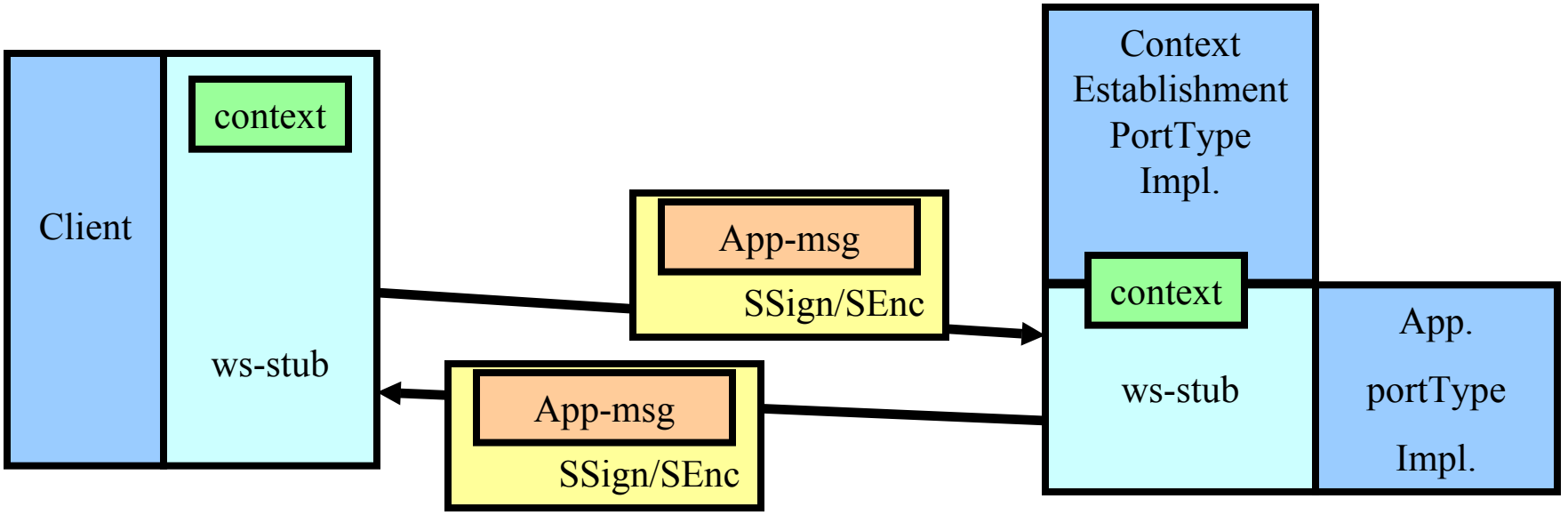
  ("tunneled" through the routing elements)

GT3 Secure Conversation: Context Establishment



- **New security context is established if none exists**
- **Dedicated context establishment portType**
- **Transparent from client and service application**

GT3 Secure Conversation: Message Protection



- **Application messages protection through established context**
- **Integrity and confidentiality protection through shared session key**
- **Transparent from client and service application**

GT3 Secure Conversation

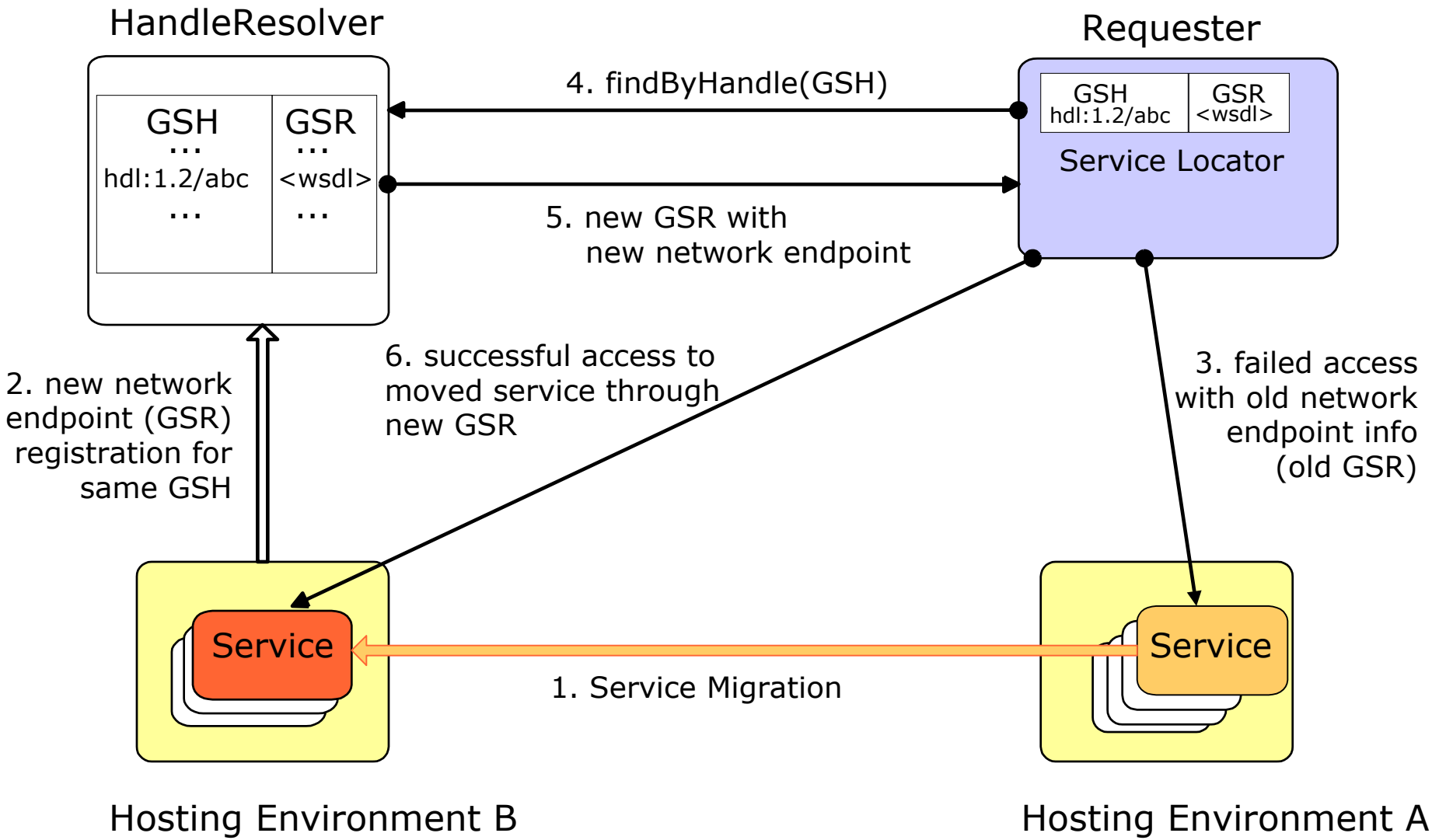
- Based on GT2's TLS/GGSAPI implementation
- Based on a poor-man's "interpretation" of WS-Trust/WS-SecureConversation specs plus XML-Signature/XML-Encryption/WS-Security

- Waiting for revised WS-Trust & WS-SecureConversation specs to be submitted to standards body
- Need a standardized message-layer, session-based authentication and key-exchange protocol
  - Maybe a GGSAPI-like equivalent, based on WS-Trust/WS-SecureConversation/XML-Signature/ XML-Encryption/WS-Security ?
- Work in GGF's OGSA-Security on hold…

OGSI and Handle Resolution

- Grid Service Handle (GSH)
  - Permanent network pointer to a Grid service
  - URI scheme indicates resolution mechanism
- Grid Service Reference (GSR)
  - Network endpoint info to access the service
  - Binding-specific (for SOAP, GSR=WSDL doc)
- HandleResolver::findByHandle
  - Service portType to resolve GSH => GSR
- Service Locator structure
  - Includes service GSHs, GSRs and portTypes
  - Factory/Find communicate Locators
- Enables transparent fail-over, load-balancing, (re-) activation, instance migration, moving services, etc.

Service Migration

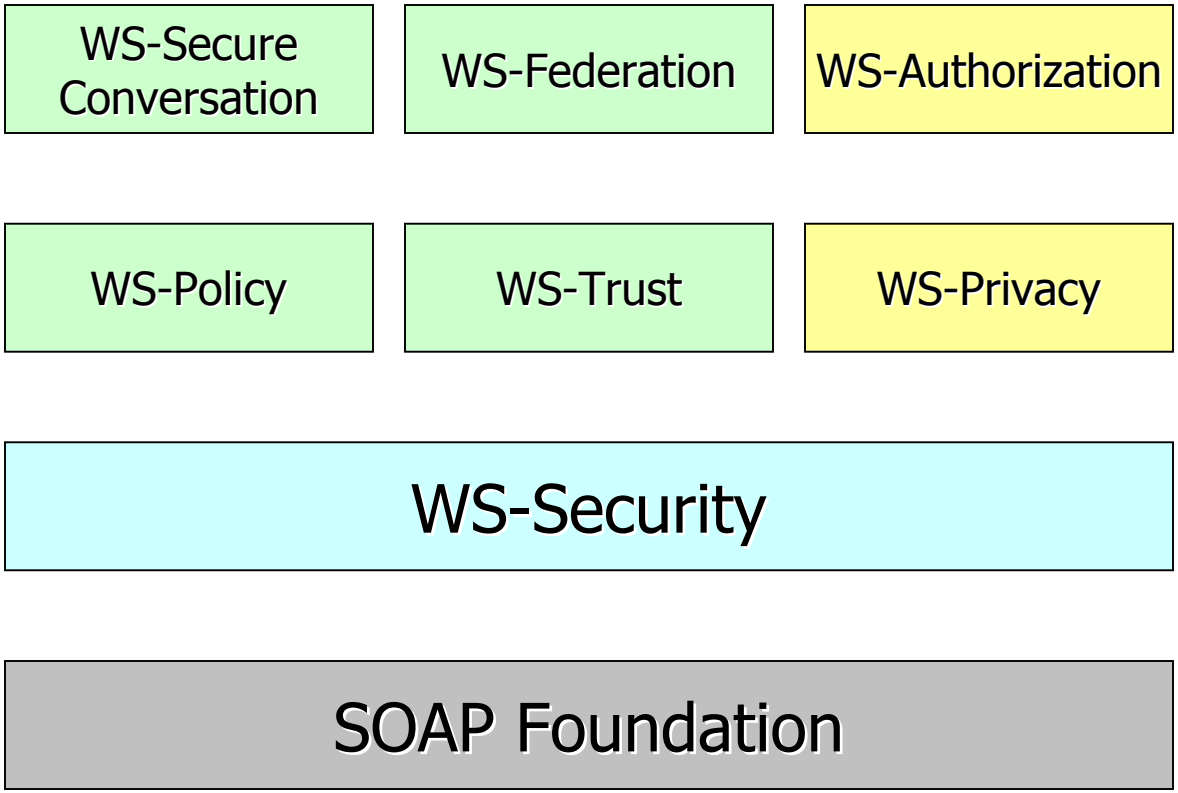


Service Instance Migration and Security

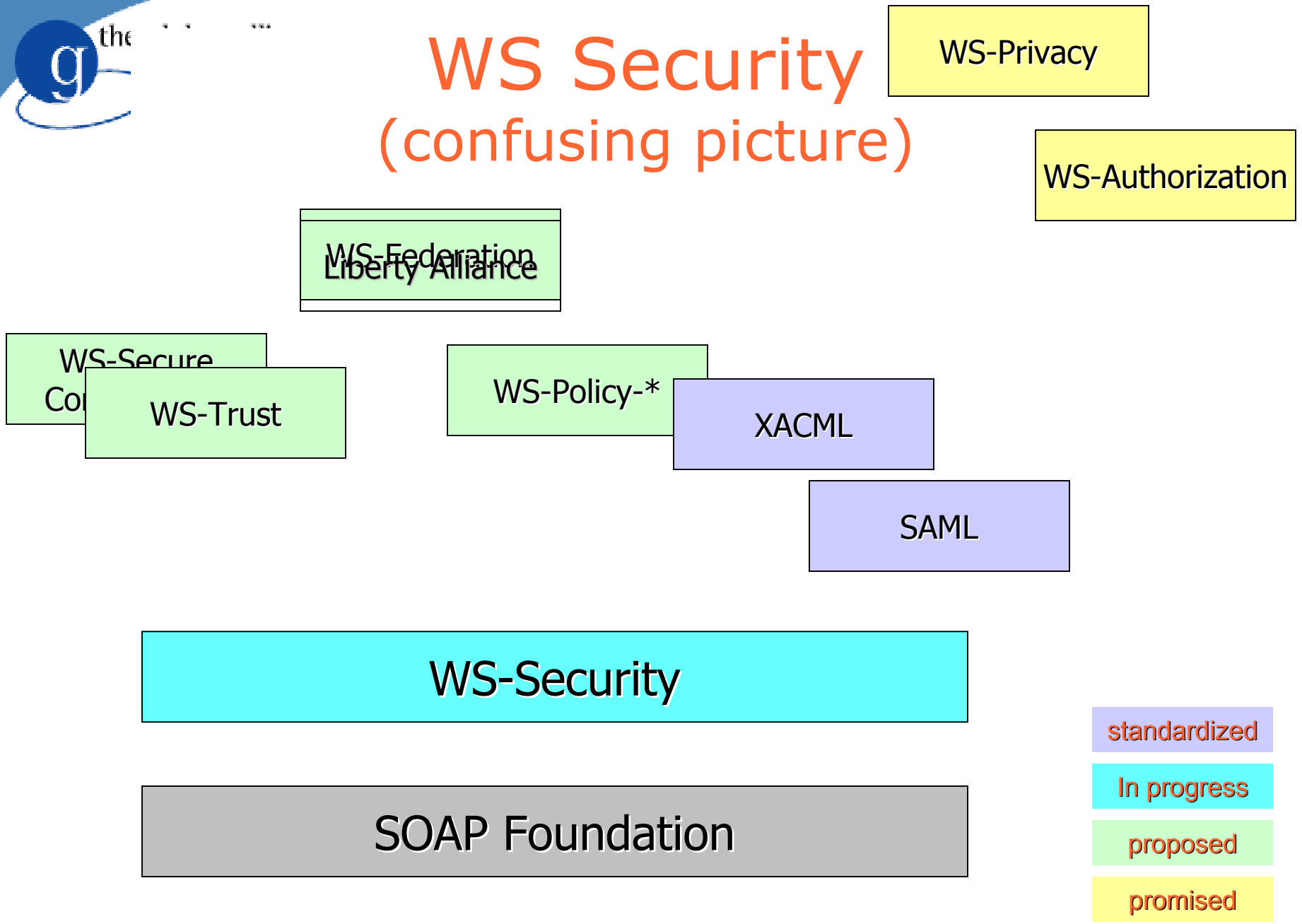
- Identity/Key "normally" associated with hosting environment and not with Instance
  - Moving instance => change of secure identity
- What about policies for that instance?
  - Users that were allowed to access, can they still access moved instance?
  - Hosting environment able to override (?)
- Where to maintain policy info?
  - Maybe in same naming/registry svc?
  - Move with instance state?
- Need more real-world requirements…
  - Learn from mobile agent systems…
  - No "real" efforts yet at GGF.

Standards and Concerns

WS Security
Current/proposed WSS-specs



WS Security (confusing picture)



Concerns about XML Security Specs (1)

- Slooow submission & standardization of specs
  - publish some specs, freeze the industry, and wait, wait, wait… until momentum is lost (?)
- IP and RF and RAND
  - Positive: most wss specs are submitted as RF
  - Clarifications take too long
  - Too many vendors involved with different T&Cs
  - Maybe authoring companies synchronize their lawyers and have single contracts…

Concerns about XML Security Specs (2)

- Interoperability
  - WS-I: Hundred+ companies, hundreds of features with tens of implementations
  - A permutation matrix nightmare…
    - But we really have to interoperate only with Microsoft's…

  Alternative:

  - Open Source Reference Implementations
    - One from Microsoft and one from IBM
      - (so we can finally help MS to debug their security code ;-)
    - Saves enormous amount of money, time, agony, travel, meetings, money, lawyers, paper, bits, bandwidth, money…
    - There is no money in plumbing anyway (as it will end up in the OS … anyway)
    - All can concentrate on the added value on top

Conclusion

- Grid's requirements maybe few years ahead, but industry will face same challenges soon
  - Few "new" distributed computing requirements…
- Our security requirements are conceptually 1-2 levels above what is available now as specifications, standards and open source
  - Ideally, we want to be end-users of wss not plumbers…
- The standards circus is very worrisome
  - And distracting and time consuming…
- Come help us at the Global Grid Forum
  - Exciting security stuff!
  - We need you… (www.ggf.org)
- Play with the "secure" new Globus Toolkit (GT3)
  - Downloaded 100k+ times already (www.globus.org)